Системы технического зрения

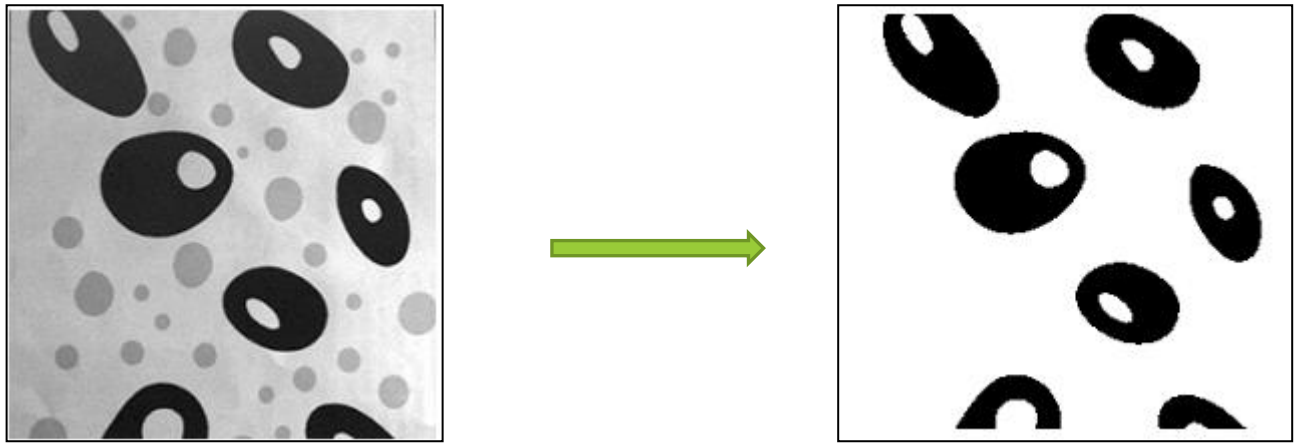
БИНАРИЗАЦИЯ. ЛИНЕЙНЫЕ ФИЛЬТРЫ

Бинарное изображение

Уменьшение объёма информации

Упрощение её обработки

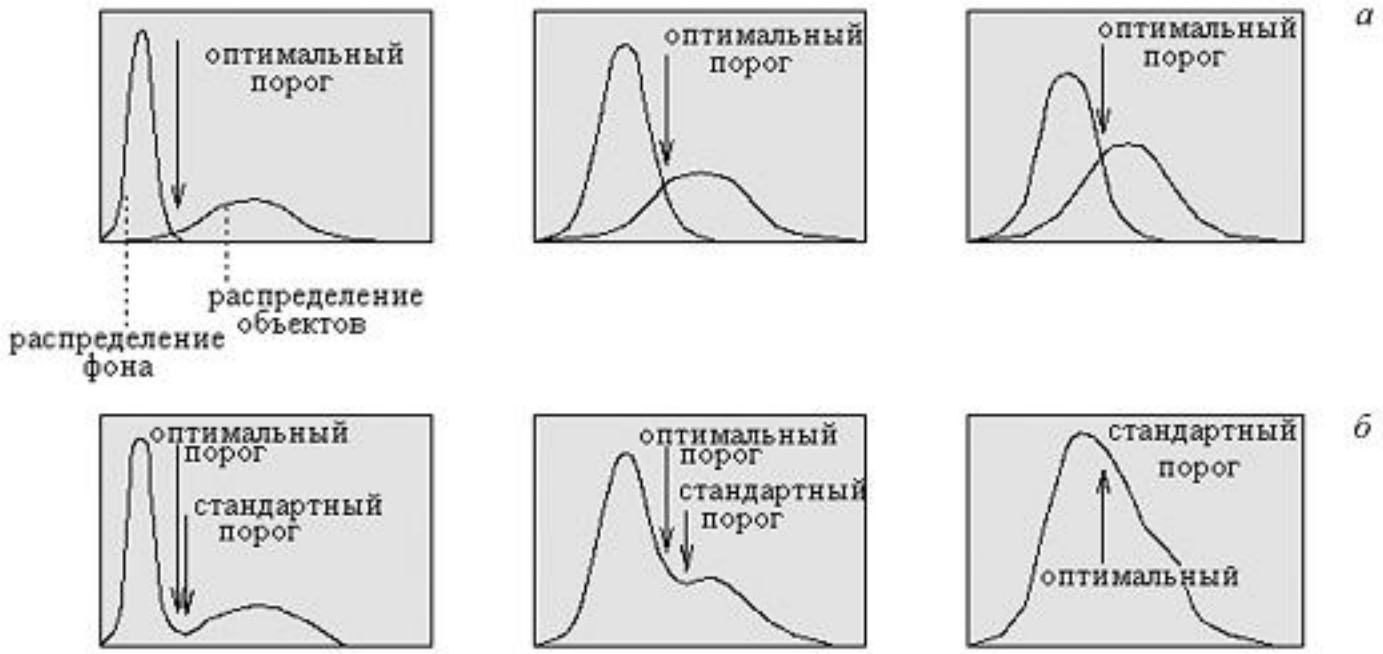
Как её произвести?



Бинарное изображение

На основе обработки гистограмы яркостей

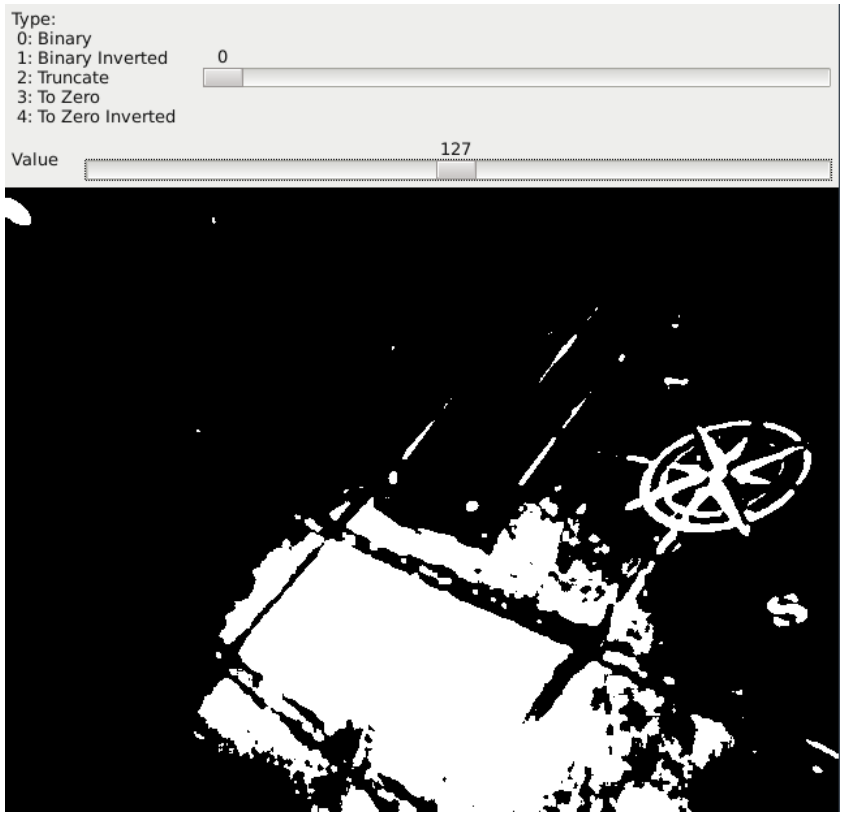
Выбор конкретного положения?



opencv_binarization

ОРИГИНАЛ

РЕЗУЛЬТАТ



opencv_binarization

```
#include <iostream>
#include <opencv2/opencv.hpp>

#include <stdlib.h>
#include <stdio.h>

using namespace cv;

/// Global variables

int threshold_value = 0;
int threshold_type = 3;;
int const max_value = 255;
int const max_type = 4;
int const max_BINARY_value = 255;

Mat src, src_gray, dst;
char* window_name = "Threshold Demo";

char* trackbar_type = "Type: \n 0: Binary \n 1: Binary Inverted \n 2: Truncate \n 3: To Zero \n 4: To Zero Inverted";
char* trackbar_value = "Value";

/// Function headers
void Threshold_Demo( int, void* );
cv::Mat brightnessEnch (cv::Mat& greyScale);
```

opencv_binarization

```
int main( )
{
  /// Load an image
  src = imread( "Iznos_sample.jpg", 1 );

  /// Convert the image to Gray
  cvtColor( src, src_gray, CV_BGR2GRAY );
  src_gray = brightnessEnch(src_gray);

  imshow ("Original",src_gray);

  /// Create a window to display results
  namedWindow( window_name, CV_WINDOW_AUTOSIZE );

  /// Create Trackbar to choose type of Threshold
  createTrackbar( trackbar_type,
                  window_name, &threshold_type,
                  max_type, Threshold_Demo );

  createTrackbar( trackbar_value,
                  window_name, &threshold_value,
                  max_value, Threshold_Demo );

  /// Call the function to initialize
  Threshold_Demo( 0, 0 );

  /// Wait until user finishes program
  while(true)
  {
    int c;
    c = waitKey( 20 );
    if( (char)c == 27 )
      { break; }
   }

}
```

opencv_binarization

```
/**
 * @function Threshold_Demo
 */
void Threshold_Demo( int, void* )
{
  /* 0: Binary                       cv::THRESH_BINARY
     1: Binary Inverted              cv::THRESH_BINARY_INV
     2: Threshold Truncated          cv::THRESH_TRUNC
     3: Threshold to Zero            cv::THRESH_TOZERO
     4: Threshold to Zero Inverted   cv::THRESH_TOZERO_INV
   */

  threshold( src_gray, dst, threshold_value, max_BINARY_value,threshold_type );

  imshow( window_name, dst );
}
```

Бинарное изображение

Критерий Отсу (Оцу)

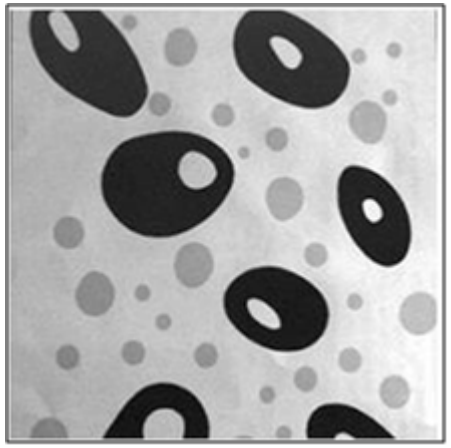
Вычисляются математицеское ожидание μ и вероятность ω для каждого значения яркости

Вычисляются их суммы и ищется минимум

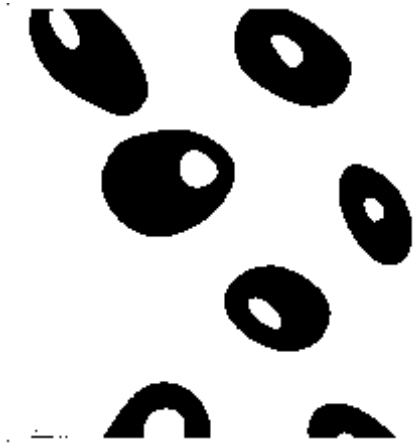
$$\sigma_b^2(t) = \sigma^2 - \sigma_w^2(t) = \omega_1(t)\omega_2(t)\left[\mu_1(t) - \mu_2(t)\right]^2$$

Бинарное изображение

ОРИГИНАЛ

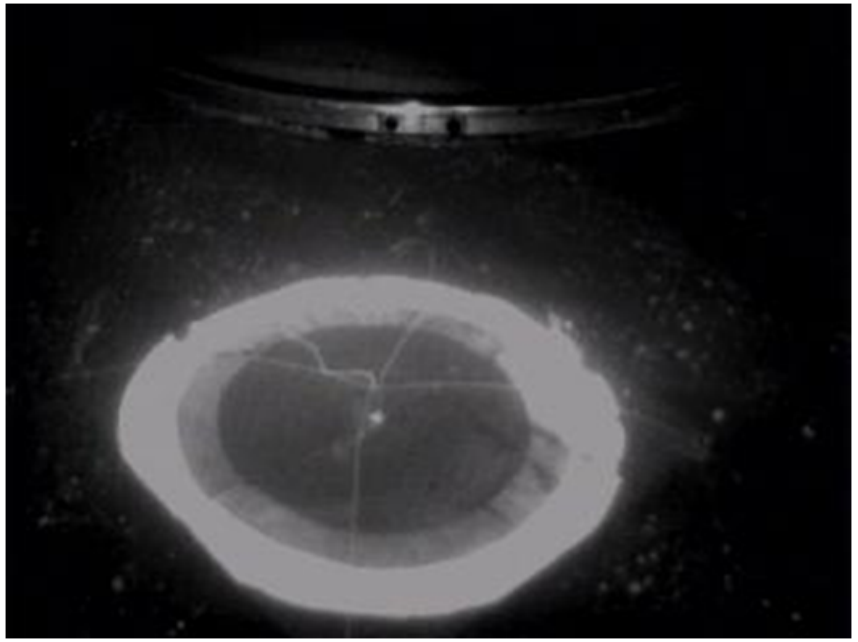


БИНАРИЗАЦИЯ THRESH_BINARY



Бинарное изображение

ОРИГИНАЛ

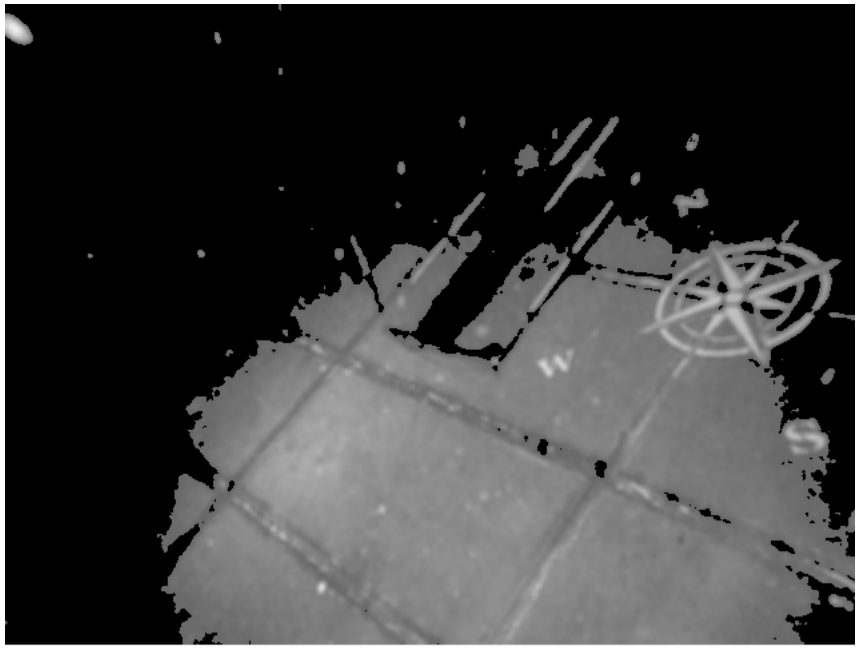


БИНАРИЗАЦИЯ THRESH_BINARY

Бинарное изображение: а если контрастность ниже?

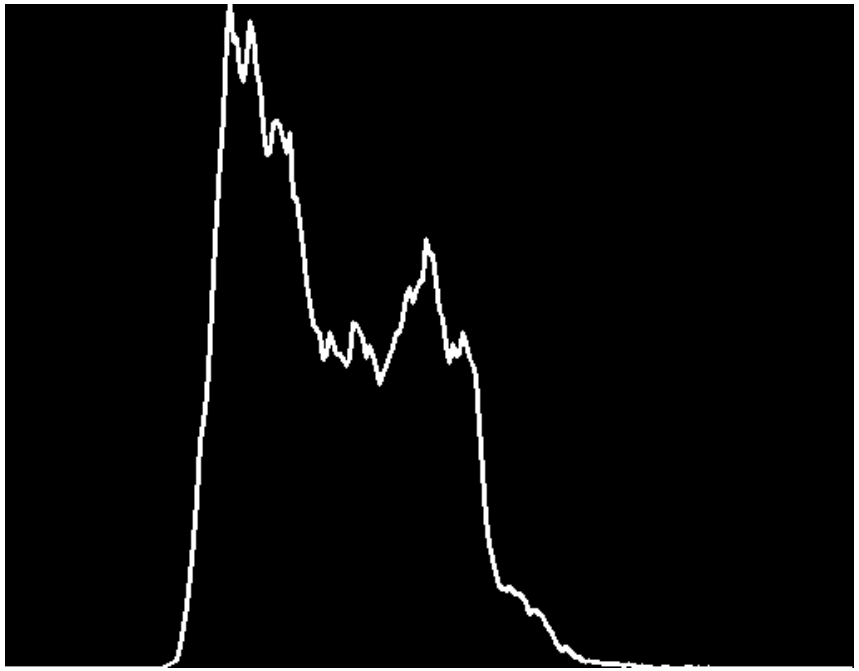
ОРИГИНАЛ

БИНАРИЗАЦИЯ TO_ZERO

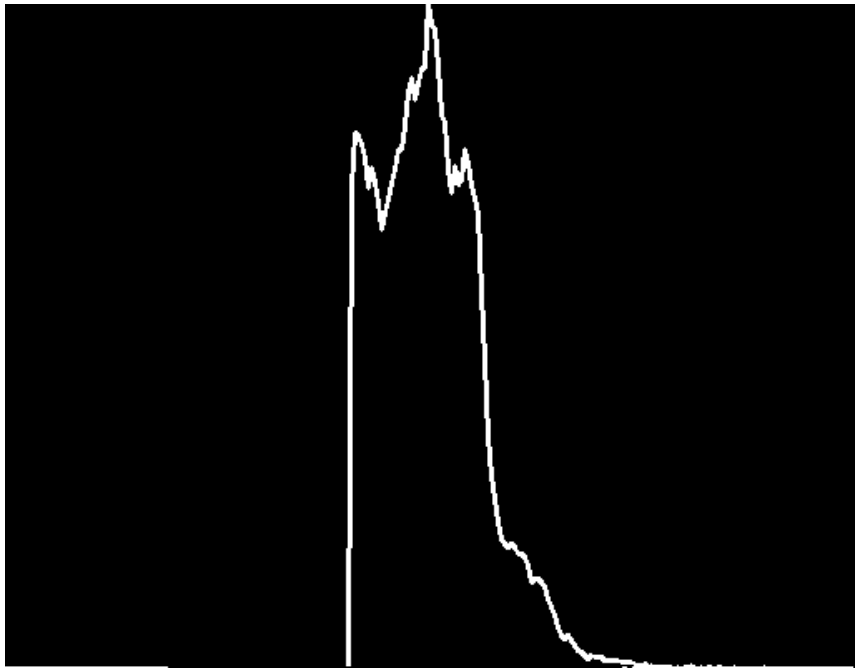


Бинарное изображение: а если контрастность ниже?

ОРИГИНАЛ



БИНАРИЗАЦИЯ TO_ZERO



Адаптивная бинаризация

Попробуем бинаризовать не всё изображение, а его отдельные области

Точнее, будем сравнивать пиксель с его окружением

Если он ярче порогового значения – 1, иначе – 0

Как выбрать область?

Адаптивная бинаризация

MEAN 5X5

1	1	1	1	1
1	1	1	1	1
1	1	1	1	1
1	1	1	1	1
1	1	1	1	1

GAUSS 5X5

1	2	3	2	1
2	4	6	4	2
3	6	9	6	3
2	4	6	4	2
1	2	3	2	1

Адаптивная бинаризация

Суммируем по области размером (size x size)

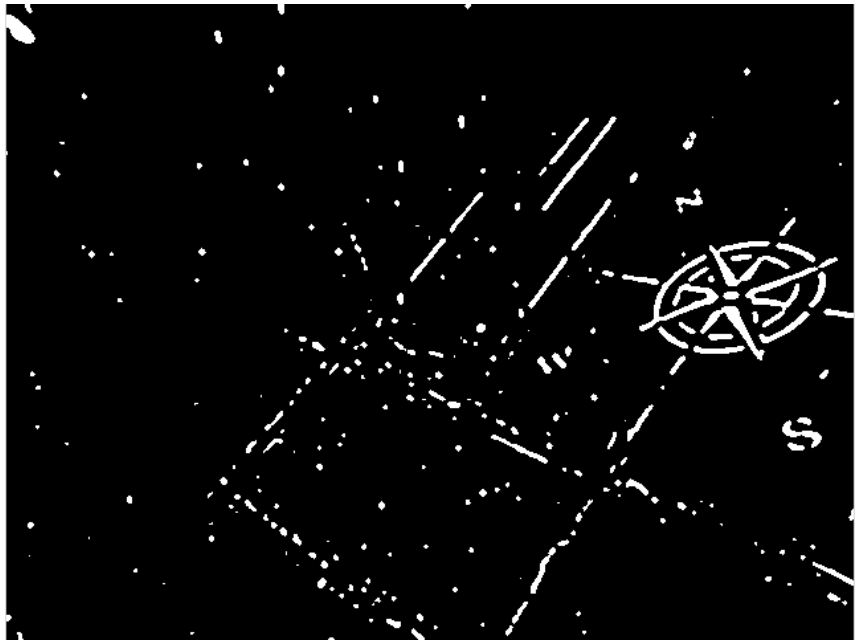
При суммировании используем веса

Делим на сумму весов

Адаптивная бинаризация

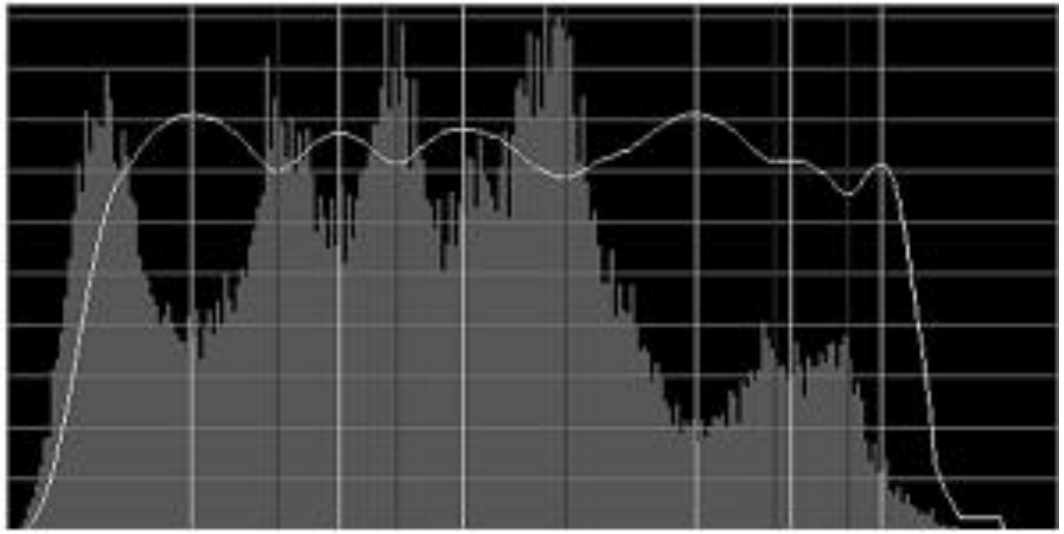
ОРИГИНАЛ

БИНАРИЗАЦИЯ MEAN



Сегментированное изображение

Для многомодового случая нет универсального подхода



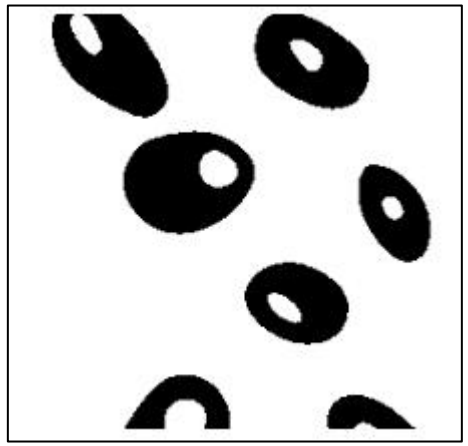
Сегментированное изображение

Бинарное изображение

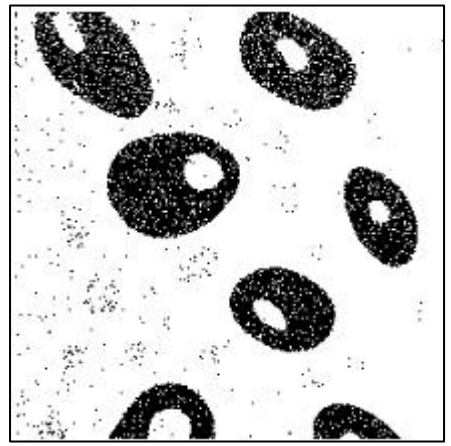
Модель шума «соль-перец»
вероятности перехода

$Im[x,y] \rightarrow Im'[x,y]$	$Im'[x,y]=1$	$Im'[x,y]=0$
$Im[x,y]=1$	$1-p$	p
$Im[x,y]=0$	q	$1-q$

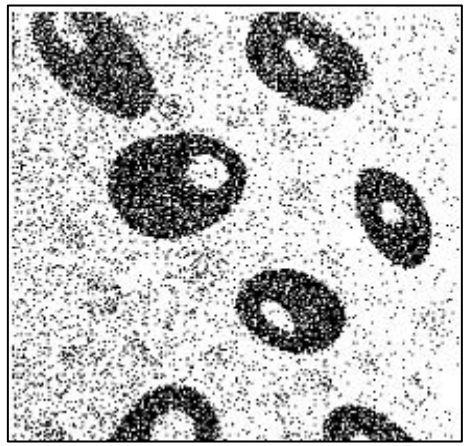
Бинарное изображение



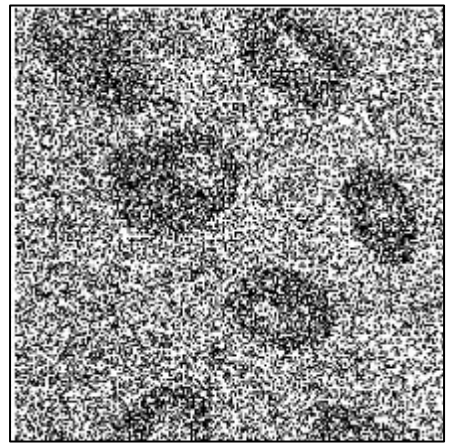
исходное



p=0.1
q=0.1



p=0.25
q=0.25



p=0.45
q=0.45

Бинарное изображение

Медианный фильтр

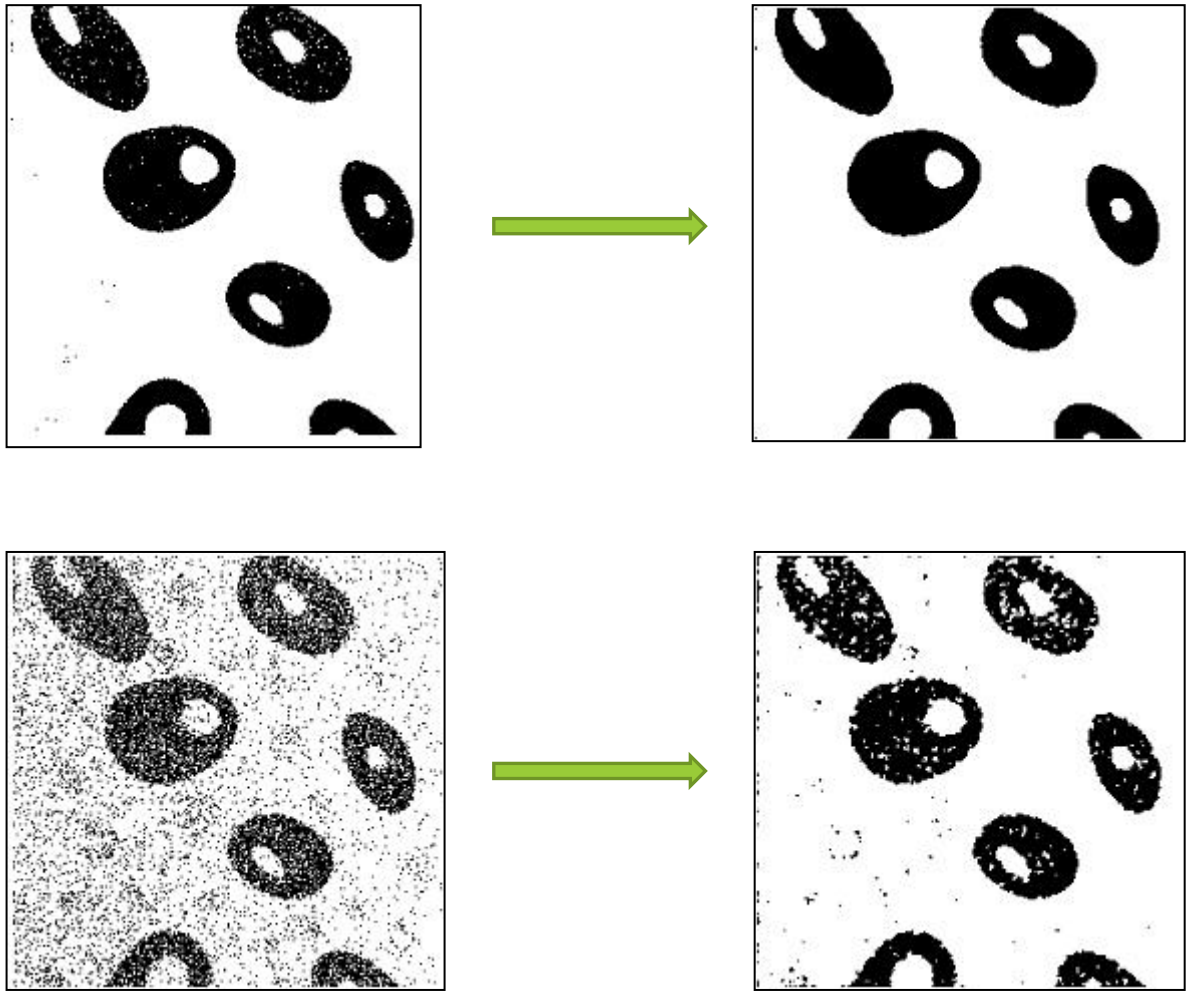
Считаем количество 0 и 1 в *n*-окрестности

1	1	0
1	0	1
1	1	0

- Единиц > нулей => 1

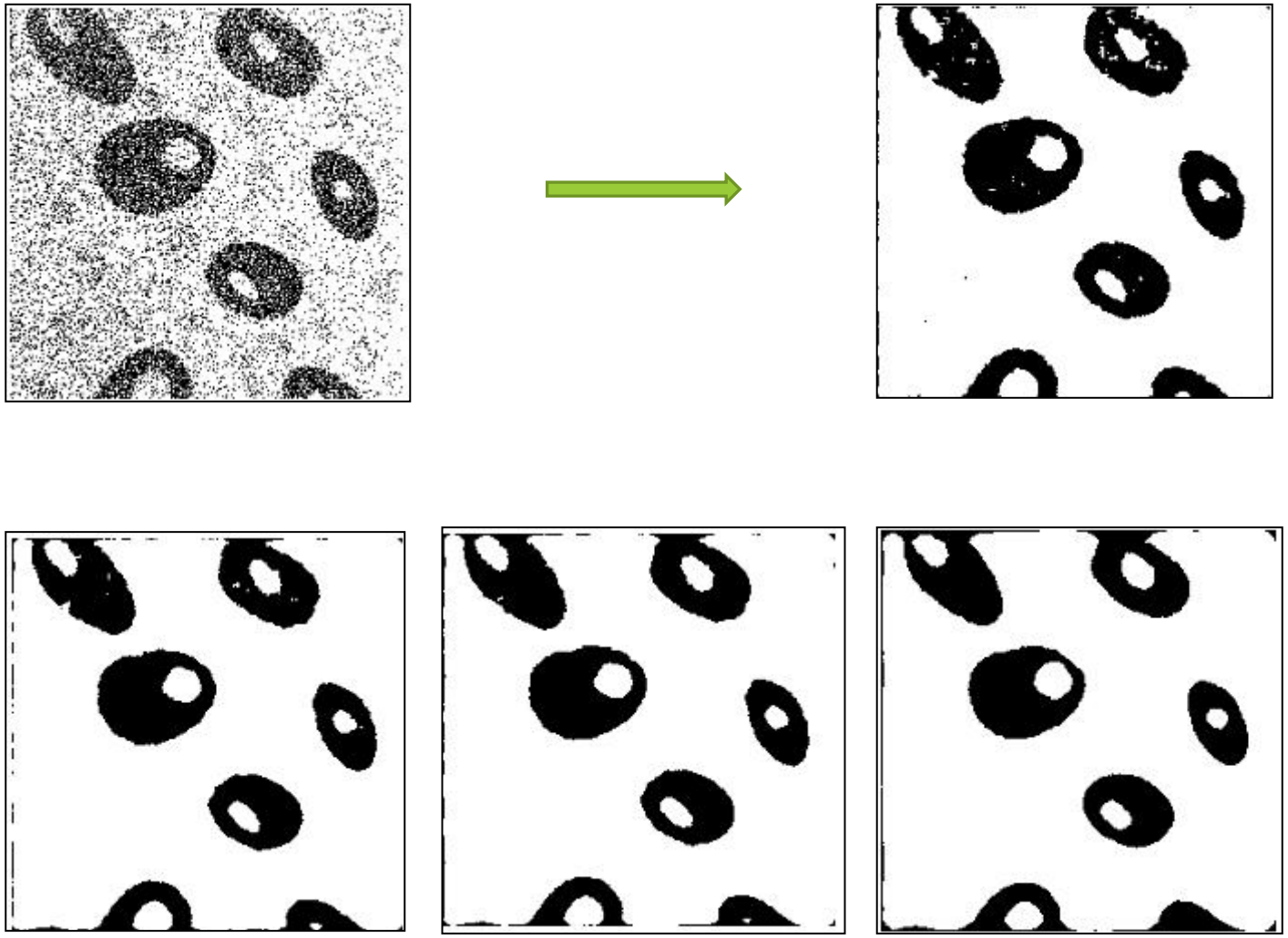
Бинарное изображение

Результат зависит от зашумлённости



Бинарное изображение

Результат зависит от размера апертуры



Бинарное изображение

ранговый фильтр – больше заданного порога

разные пороги для нулей и единиц!

Он же процентильный

1	1	0
1	0	1
1	1	0

- Единиц 6, нулей 3
- Ранг 6 – 1, ранг 7 – 0
- Нужен, если мы имеем априорную информацию

Гауссов шум

Модель аддитивного шума

$$Im'[x,y]=Im[x,y]+R(x,y)$$

Частный случай – гауссов шум

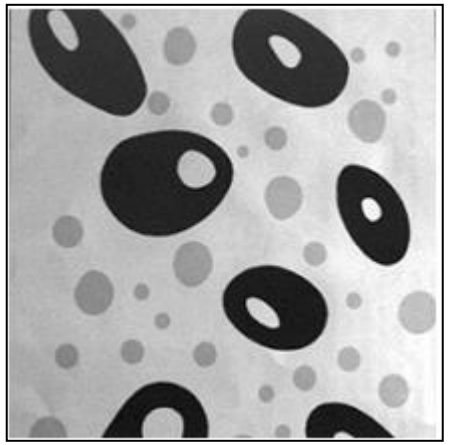
$$Im'[x,y]=Im[x,y]+N(0,\sigma),$$

Мат.модель: сумма множества независимых факторов

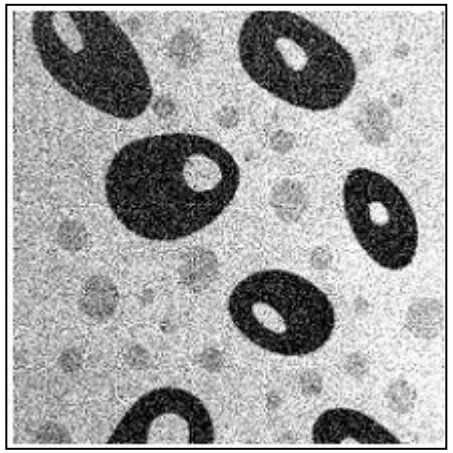
Подходит при маленьких дисперсиях

Предположения: независимость, нулевое матожидание

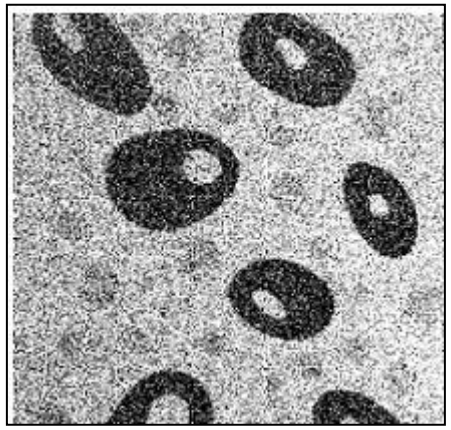
Гауссов шум



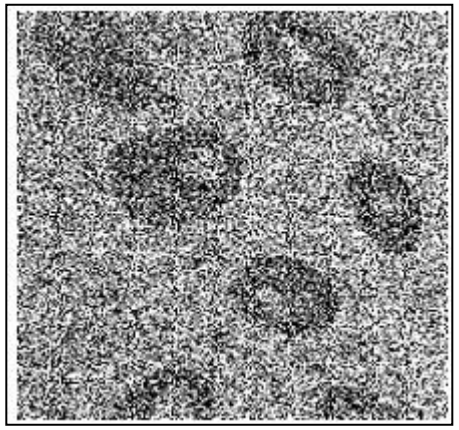
исходное



σ=40



σ=80



σ=300

Гауссов шум

Медианный фильтр

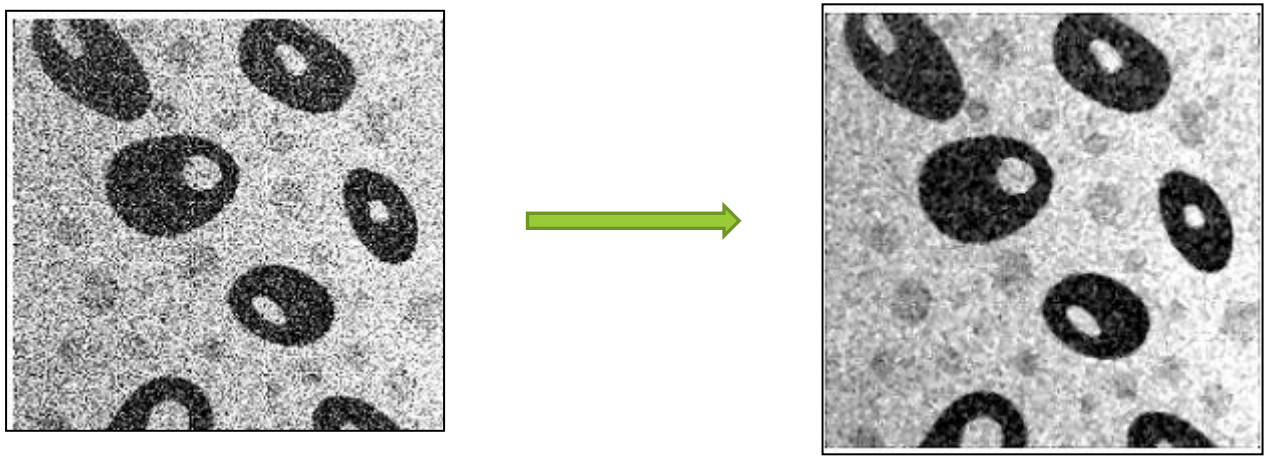
Упорядочиваем точки в *n*-окрестности

173	164	170
150	176	169
168	182	166

- (150, 164, 166, 168, 169, 170, 173, 176, 182) – в середине списка

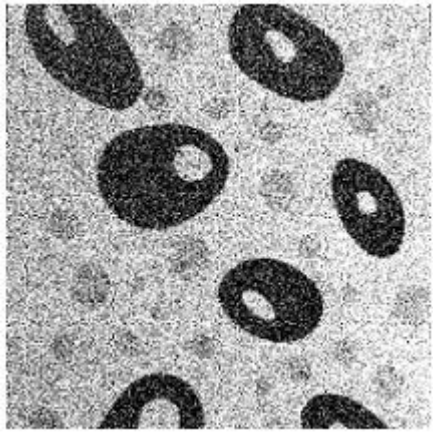
Гауссов шум

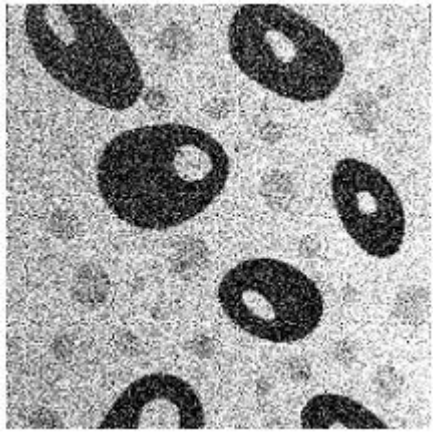
Зависит от зашумлённости

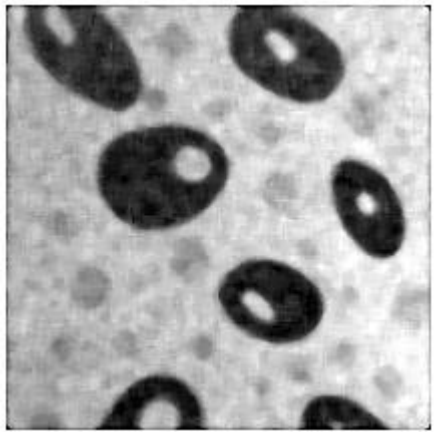


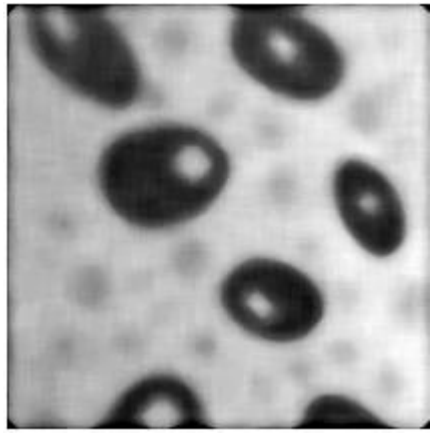
Бинарное изображение

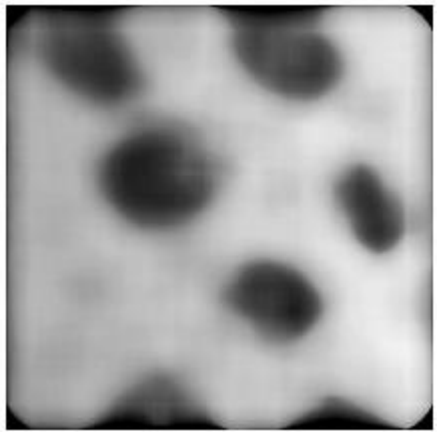
Зависит от размера апертуры











Гауссов шум

Ранговый фильтр

Упорядочиваем точки в *n*-окрестности

173	164	170
150	176	169
168	182	166

- (150, 164, 166, 168, 169, 170, 173, 176, 182) – в середине списка

Гауссов шум

Взвешенный фильтр

Суммируем с весами

173	164	170
150	176	169
168	182	166

1	1	1
1	1	1
1	1	1

- $\Sigma 167{,}33 \approx 167$

Гауссов шум

Взвешенный фильтр

Суммируем с весами

173	164	170
150	176	169
168	182	166

1	2	1
2	4	2
1	2	1

- $\Sigma 169{,}43 \approx 169$

Фильтры в OpenCV

Медианный фильтр

void medianBlur (cv::Mat src, CV::Mat dst, int ksize)

src – исходное изображение

dst – результат

ksize – размер фильтра

Бокс-фильтр

void boxFilter (cv::Mat src, CV::Mat dst, int ksize)

Гауссов фильтр

void gaussianBlur (cv::Mat src, CV::Mat dst, int ksize)

Результат фильтрации

ОРИГИНАЛ

БОКС-ФИЛЬТР

Результат фильтрации

ОРИГИНАЛ

СВЁРТКА С ГАУССИАНОМ

Результат фильтрации

ОРИГИНАЛ

МЕДИАННЫЙ ФИЛЬТР